

UNIVERSIDADE DE LISBOA

Faculdade de Ciências

Departamento de Informática



SIMULADOR DE MENSAGENS AFTN E OLDI – TRANSMISSÃO

João Miguel Carvalheira da Cruz

Relatório Público

MESTRADO EM ENGENHARIA INFORMÁTICA

Especialização em Sistemas de Informação

2010/2011

UNIVERSIDADE DE LISBOA

Faculdade de Ciências

Departamento de Informática



SIMULADOR DE MENSAGENS AFTN E OLDI – TRANSMISSÃO

João Miguel Carvalheira da Cruz

PROJECTO

Trabalho orientado pela Prof. Doutor Maria Dulce Pedroso Domingos

e co-orientado pelo Eng. Francisco José Salgado Caldeira.

MESTRADO EM ENGENHARIA INFORMÁTICA

Especialização em Sistemas de Informação

2010/2011

Agradecimentos

Como é a primeira vez que escrevo algo deste género em todo o meu editorial de vida, aproveito esta oportunidade para agradecer a todos os que contribuíram e contribuem a colocar mais um marco na minha vida.

Para ser pouco coerente com a finalidade deste registo, quero agradecer a mim próprio, porque sim, porque sem mim nada disto faria sentido e porque sim, apesar de tudo eu também mereço.

Agora que já todos perderam a esperança do merecido reconhecimento, quero que saibam que coloco este marco graças a vocês. A começar pela minha professora da escola primária que me preparou devidamente para a vida académica e me fez pensar que um dia viria a ser escritor. Peço desculpa D. Isabel, por não ter correspondido minimamente ao idealizado.

Seguindo directamente para o preparatório, quero agradecer a todos os professores que tiveram a coragem de me atribuir notas “menos boas”, especialmente à professora Elsa que me fez ver que para chegar ao topo também é preciso algum esforço e dedicação. Passado com distinção, chego à escola de todas as decisões e avaliações que me permitiram chegar a este parágrafo. Ao professor Jaime Filipe, por quem nutro grande admiração como professor e pessoa, agradeço as horas dedicadas ao ensino da matemática, contribuindo favoravelmente para o meu destino universitário.

Finalmente, após início atribulado ou falta dele, quero agradecer a todos os professores que fizeram parte do percurso universitário, por nunca me terem incentivado ao plágio e fazerem-me acreditar que Deus existe nos momentos mais difíceis.

Agradeço especialmente à professora Dulce Domingos, de quem obtive os meus primeiros 20 valores e que por coincidência é a orientadora deste estágio (só pode ser bom presságio!?).

Iniciado o estágio de Mestrado, enalteço as condições de trabalho proporcionadas pela empresa NAV Portugal que contribuíram para a qualidade do trabalho desenvolvido.

Como pessoa responsável pelo projecto de estágio e orientador quero agradecer ao Dr. Rui Azedo por me ter proposto a realização deste projecto exigente e prestigiante. Como

também pelos conselhos dados e empenho prestado através do seu conhecimento e experiência.

Como co-orientador de estágio, agradeço ao Engº. Francisco Caldeira pelo tempo dedicado (de qualidade) ao desenvolvimento do projecto, mostrando-me sempre o outro lado do problema ou um novo problema (*testing capability*). Especial agradecimento à Engª. Maria José Serrano pelo apoio dado desde do início do projecto e simpatia demonstrada.

A todos os colegas da DSTI e colegas de estágio que trabalharam, almoçaram e conviveram comigo, um grande obrigado. Destaque especial ao colega João Manuel Martins por me ter recebido gentilmente no seu gabinete e nos garantir uma fatia do jackpot do euromilhões mesmo sem nunca termos contribuído. Fica registado, que até ao momento ainda não se encontrou ursos ou mesmo “nada”, no barrocal alentejano.

Agradeço ao J.Tiago Reis a parceria e companheirismo demonstrados ao longo do estágio, com a certeza que os simuladores de mensagens AFTN e OLDI nunca mais serão os mesmos a partir de agora.

Por fim, não podia deixar de referir toda a minha família que me acompanha e dá força em cada novo desafio da minha vida. Sabendo que teriam de ser muitas as palavras para exprimir a minha eterna gratidão, à minha mãe Mª Fernanda, ao pai César, à mana Sara Isabel, à minha Ana e ao meu pequenote Miguel Filipe, Muito Obrigado.

Dedico,

Aos meus pais, irmã, à minha Ana e filho Miguel Filipe

Resumo

O projecto realizado na empresa NAV Portugal EPE enquadra-se na actividade de navegação aérea e consiste no desenvolvimento de uma aplicação que permite simular a transmissão de mensagens segundo os protocolos AFTN¹ e OLDI² para um sistema de gestão de tráfego aéreo.

A empresa é responsável pelo controlo e gestão de tráfego aéreo (ATM³) em Portugal (Lisboa e Santa Maria FIR) e pelos serviços de navegação nos principais aeroportos de Portugal. O controlo e gestão de tráfego aéreo são compostos pela integração funcional de elementos terrestres e aéreos. Em terra, os controladores situados nos centros de controlo e tráfego aéreo monitorizam e orientam aeronaves no ar e no solo, de forma a assegurar um fluxo de tráfego seguro, ordenado e expedito. O sistema de gestão de tráfego aéreo - LISATM⁵ oferece mecanismos de controlo e gestão de informação em tempo-real aos controladores para auxílio das suas funções. O sistema é composto por diversos meios técnicos (comunicações, radares, rádio, computadores, etc.) que agregam informação de dados de plano de voo e dados de vigilância. No ar, o operador da aeronave cumpre a rota definida através de um plano de voo anteriormente submetido à entidade CFMU¹¹ que regula o tráfego aéreo europeu, o qual pode ser alterado conforme os congestionamentos. Os planos de voo chegam ao sistema LISATM através de mensagens AFTN onde são processadas e extraídos os dados relevantes para a FIR¹² de Lisboa.

Para além do comando das aeronaves, os operadores têm a tarefa de seguir as directrizes do controlador, de acordo com o cenário corrente. Para os voos que entram na FIR de Lisboa, o órgão responsável pela FIR adjacente notifica e coordena com o ACC⁴ de Lisboa as condições de entrada no espaço aéreo. O sistema LISATM, interligado com os sistemas das FIR adjacentes, garante a coordenação, notificação e revisão das condições de entrada dos voos através de mensagens trocadas entre sistemas, designadas mensagens OLDI. A informação actualizada no sistema LISATM

através das mensagens OLDI permite obter com maior rigor os dados da aeronave em vias de entrar na área de responsabilidade do ACC.

Para fazer face aos requisitos operacionais do sistema LISATM é utilizado um sistema (Pre-OJT⁹) de testes e validação de funcionalidades. Este sistema é utilizado para testar novos procedimentos operacionais e para treino dos controladores antes da entrada em operação de novas funcionalidades. A arquitectura do sistema Pre-OJT integra o bloco de simulação SIMATM⁸ e o bloco de treino. O bloco de simulação é um sistema que recria eventos do mundo real através de dados gerados a partir de cenários de tráfego aéreo previamente definidos. O bloco de treino consiste numa réplica do sistema LISATM, o qual recebe dados do bloco de simulação.

Actualmente, os testes ao sistema com mensagens do tipo AFTN e OLDI são efectuados através da injeção de scripts no sistema LISATM. Os scripts são escritos manualmente. Para efectuar testes de funcionalidades do sistema existe a necessidade de adaptar e reutilizar eventos de acordo com o momento temporal em que se desenrolam. Este processo manual sendo moroso, sobrecarrega os operadores para além do tempo desejável para execução de testes de funcionalidades.

Neste projecto foi desenvolvido um sistema aplicacional que oferece uma melhoria efectiva do método utilizado por scripts. A aplicação permite a injeção, criação e gestão de informação das mensagens dos protocolos AFTN e OLDI de modo a tornar mais fácil, rápido e automáticos os testes de funcionalidades do sistema LISATM.

Palavras-chave: ATM, AFTN, OLDI, SIMATM

Abstract

The project fits into business or air navigation. Undertaken at NAV Portugal EPE it consisted of developing an application that allows you to simulate the transmission of messages according to the protocols AFTN and OLDI to a system of air traffic management. NAV Portugal EPE have its business activity on monitoring and managing air traffic control (ATC / ATM) in Portugal (Lisbon and Santa Maria FIR) and the navigation services at major airports in Portugal. The air traffic management is formed by the functional integration of ground and air elements. On ground, the controllers located in air traffic control centers, monitor and guide aircraft in the air/ground to ensure a safe traffic flow, orderly and expeditious. A system of air traffic management (LISATM) formed by various technical means (communications, radar, radio, computers, etc.) aggregates data information from the flight plan and surveillance data. The system provides control and information management mechanisms in real-time to help out controllers through their duties. In air, the aircraft operator meets the defined route through a flight plan previously submitted to the entity that governs the European air traffic, CFMU, which can be changed according through traffic jams. Flight plans arrive to LISATM system by AFTN messages which are processed, extracting relevant data to the Lisbon FIR. Beyond the aircraft control, operators have the task of following the controllers guidelines according to the current scenario. For flights arriving the Lisbon FIR, the agency responsible for that adjacent FIR notifies and coordinates with the Lisbon ACC the conditions for entering the airspace of Lisbon FIR. The LISATM system interconnected with the systems of the adjacent FIR ensures coordination, notification and review of the entry conditions of new flights through messages exchanged between systems, designated OLDI messages. Updated information on the system through OLDI messages offers accurate data from the aircraft about to enter the area of the responsibility of the ACC. To meet the operational needs of LISATM system is used a Pre-OJT system for testing and validation of features. The system is used to test new operational procedures and training controllers before the entry of new features. The Pre-OJT system architecture integrates block of simulation – SIMATM and the block ok training. The block of simulation is a system that recreates real-world events, using data generated from air traffic scenarios previously defined.

The block of training is a replica of LISATM system which receives data from simulation block.

Currently testing the system with AFTN and OLDI messages are done by injecting scripts into the system LISATM. The scripts are written manually. For testing of functionalities of the system there is the need to adapt and reuse events according to the time frame in which they unfold. The use of manual process becomes time consuming and hard on the operators for beyond the desirable time for the test of application functionalities. In this project was developed an application system that provides an effective improvement of the method used by scripts. The application allows the injection, creation and management of AFTN and OLDI protocol messages for the test of functionalities of the LISATM system.

Keywords: ATM, AFTN, OLDI, SIMATM

Conteúdo

Capítulo 1	Introdução.....	1
1.1	Empresa	1
1.2	Integração Profissional	2
1.3	Controlo e Gestão de Tráfego Aéreo	2
1.4	Simulador ATM System.....	3
1.5	Motivação	4
1.6	Objectivos.....	5
1.7	Organização do documento	6
Capítulo 2	Contexto	7
2.1	Sistema LISATM.....	7
2.2	Navegação	7
2.3	Aeronautical Fixed Telecommunications Network.....	8
2.4	On-line Data Interchange.....	10
Capítulo 3	Metodologia e Planeamento	13
3.1	Metodologia.....	13
3.2	LISATM	14
3.3	Processo de Desenvolvimento	14
3.4	Planeamento do Projecto	15
3.4.1	Desvios.....	19
Capítulo 4	Desenvolvimento do Projecto	20
4.1	Análise de Requisitos	20
4.2	Desenho	20
4.2.1	Padrões de Desenho	20
4.2.2	Modelo de Classes	22
4.3	Implementação.....	22
4.3.1	Modelo de Dados	23
4.3.2	Estrutura do Código	23

4.3.3	Comunicação.....	23
4.3.4	Interface de Utilizador	23
4.4	Testes.....	30
4.4.1	Modelo de Testes	31
Capítulo 5	Conclusão.....	32
Bibliografia		34
Glossário		37

Lista de Figuras

Figura 1. Arquitectura Pre-OJT/Arquitectura do sistema de testes LISATM.....	4
Figura 2. Estrutura operacional do CFMU.	8
Figura 3. Exemplo de processo de transferência de controlo de voo para unidade ATC adjacente – LPPT.	12
Figura 4. Exemplos de notação UML.	13
Figura 5. Ciclo de vida do Processo Unificado de Desenvolvimento de Software	15
Figura 6. Mapa de Gantt referente ao planeamento do projecto	18
Figura 7. Estrutura do padrão <i>Model-View-Controller</i> para uma aplicação <i>GUI</i> ...	21
Figura 8. Interface de utilizador – Autenticar utilizador.....	25
Figura 9. Gestor de cenários – Seleccionar cenário; Introduzir nome de cenário ..	26
Figura 10. Gestor de cenários – Editar cenário; Copiar cenário, Apagar cenário...	26
Figura 11. Gestor de voos – Seleccionar voo; Criar novo voo	27
Figura 12. Gestor de Voos – Editar voo; Copiar voo, Apagar voo; Menu <i>Pull Down</i>	27
Figura 13. Gestor de Mensagens – Seleccionar mensagem, Criar nova mensagem	28
Figura 14. Gestor de Mensagens – Editar mensagem; Validar mensagem; Ver Sintaxe	28
Figura 15. Simulador – Gerir mensagens do exercício.....	29
Figura 16. Simulador – Iniciar exercício	30

Lista de Tabelas

Tabela 1. Planeamento geral de trabalho	16
Tabela 2. Planeamento detalhado de trabalho.....	17

Capítulo 1

Introdução

Este relatório descreve o projecto de engenharia informática realizado na empresa NAV Portugal E.P.E. – Navegação Aérea de Portugal, no âmbito do Mestrado em Engenharia Informática da Faculdade de Ciências da Universidade de Lisboa. O projecto consiste no desenvolvimento de uma aplicação que permita simular a transmissão de mensagens segundo os protocolos **AFTN**¹ e **OLDI**² para um sistema de gestão de tráfego aéreo.

1.1 Empresa

A evolução tecnológica como factor de inovação tem contribuído para a dinamização e vanguarda do negócio no sector da navegação aérea.

A empresa de navegação aérea em Portugal (NAV Portugal) é o espelho do crescimento sustentado através da introdução de tecnologias actuais que têm permitido dinamizar o controlo e gestão de tráfego aéreo, ao integrar segurança e eficiência, premissas essenciais ao processo de controlo de tráfego aéreo e sua melhoria.

A NAV Portugal E.P.E. é responsável pelo controlo e gestão de tráfego aéreo no espaço aéreo sob a responsabilidade de Portugal (Portugal Continental, Regiões Autónomas e vasta área sobre Atlântico Norte) e serviços de navegação nos principais aeroportos de Portugal.

¹ AFTN – Aeronautical Fixed Telecommunications Network

² OLDI – On-line Data Interchange

1.2 Integração Profissional

O projecto foi desenvolvido na sub-área SISINT – Sistemas Interface com o Utilizador que faz parte da estrutura da DSTI - Direcção de Sistemas e Tecnologias de Informação da NAV Portugal.

A estrutura da DSTI - Direcção de Sistemas e Tecnologias de Informação possui as seguintes sub-áreas:

- SISINT – Sistemas, Interface com o Utilizador
- SISPRO – Sistemas e Software, Produção
- SISQUA – Gestão da Qualidade dos Sistemas
- SISLOG – Sistemas, Logística

Na fase inicial do trabalho tomei conhecimento da estrutura e negócio da empresa, as funções de cada sub-área da DSTI e métodos de trabalho utilizados no percurso de desenvolvimento de um projecto.

Este projecto durante o seu ciclo de vida teve o apoio de todas as sub-áreas.

Durante o período de integração na empresa, como complemento realizei uma actividade de formação enquadrada na área do projecto a realizar.

1.3 Controlo e Gestão de Tráfego Aéreo

A gestão de tráfego aéreo – ATM³ é composta pela integração funcional de elementos terrestres e aéreos durante as operações de navegação. Os controladores aéreos situados em centros de controlo de gestão de tráfego aéreo - ACC⁴, monitorizam e orientam aeronaves no ar e no solo, de forma a assegurar um fluxo de tráfego seguro, ordenado e expedito. A informação gerada entre os vários elementos do processo de navegação flui através de um sistema de gestão de tráfego aéreo.

³ ATM – Air Traffic Management

⁴ ACC – Air Control Center

O sistema de gestão implementado na NAV Portugal desde 2001 está integrado no centro de controlo de tráfego aéreo de Lisboa - **LISATM**⁵ e nas torres de controlo do continente e Madeira - **TWRATM**⁶.

O sistema integra uma rede local de alta disponibilidade redundante, que garante a estabilidade operacional e segurança da navegação.

O sistema LISATM é composto por dois sistemas, um sistema operacional e outro *fallback* do anterior.

O sistema operacional contempla diferentes tipos de equipamentos e subsistemas (de ajuda à navegação, vigilância, comunicações de voz e dados, computadores, etc.).

Estes subsistemas fornecem informação em tempo-real, tal como dados de vigilância (que indicam a posição das aeronaves) e dados de planos de voo (indicam a intenção do voo, aeroporto de partida, rota, aeroporto de destino e características da aeronave) que o sistema de gestão correlaciona e apresenta aos controladores de tráfego aéreo. Com base nesta informação e com o auxílio de ferramentas específicas que fornecem alertas quando as aeronaves se aproximam demasiado e avisos quando se desviam das instruções dadas pelos controladores, *SAFETY NETS*⁷, é possível garantir uma circulação segura e eficaz de aeronaves.

1.4 Simulador ATM System

O SIMATM⁸ – bloco de simulação e bloco de treino estão funcionalmente integrados na arquitectura do sistema Pre-OJT⁹ - ver Figura 1 – é usado para testar novos procedimentos operacionais e para treino dos controladores antes da entrada em operação de novas funcionalidades.

⁵ LISATM – Lisbon ATM System

⁶ TWRATM – Control Tower ATM System

⁷ SAFETY NETS – Ferramenta que fornece alertas e avisos de segurança para a prevenção de acidentes.

⁸ SIMATM – Simulator ATM System

⁹ Pre-OJT – Pre- On Job Training

O bloco de treino consiste numa réplica parcial do sistema LISATM que providencia o mesmo comportamento observado na posição de controlo - CWP¹⁰ ao receber dados de simulação do bloco de simulação.

O bloco de simulação - SIMATM é um sistema de simulação ATM de fácil adaptação aos vários sistemas da NAV Portugal que recria eventos do mundo real, através de dados gerados a partir de cenários de tráfego aéreo previamente definidos (dados radar, dados de plano de voo, etc.). É composto por um conjunto bem definido de blocos funcionais, onde cada um deles é por sua vez composto por um conjunto de módulos *pluggable*. Este bloco SIMATM (cópia parcial do sistema de gestão) quando ligado a um sistema de teste, ao testar novas versões de *software*, permite recriar situações de tráfego específicas, que seriam mais difíceis de obter e de reproduzir com tráfego real.

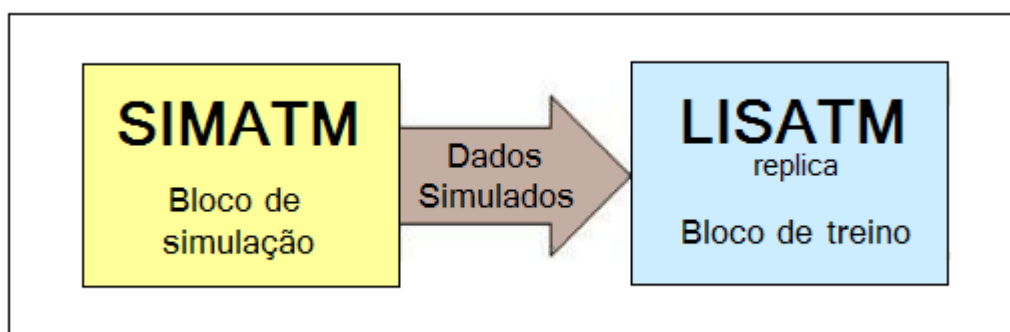


Figura 1. Arquitectura Pre-OJT/Arquitectura do sistema de testes LISATM.

1.5 Motivação

Em concordância com os compromissos internacionais relativos à harmonização e integração de sistemas, crescimento do tráfego aéreo, reforço da segurança aérea e melhoria da qualidade e eficiência do serviço é necessária a constante melhoria e actualização do processo de controlo de tráfego aéreo. Face às necessidades operacionais do sistema LISATM e TWRATM são realizadas actualizações periódicas com novas funcionalidades que são verificadas através do sistema de simulação –

¹⁰ CWP – Control Working Position

SIMATM, a partir do qual são efectuados testes sistemáticos e validação de funcionalidades para garantir a fiabilidade e segurança.

Actualmente os testes de funcionalidades que visam a transmissão de mensagens AFTN e OLDI são feitos através da injeção de *scripts* no sistema. Estes *scripts* simulam as acções dos sistemas externos, o que permite verificar o funcionamento do sistema LISATM/TWRATM.

Os *scripts* desenvolvidos contêm informação representada de acordo com a sintaxe definida pelo tipo de formato da mensagem pretendida.

A problemática deste processo reside na realização de testes de aceitação e testes formais na entrada do sistema. Quando o operador simula as mensagens através da injeção de *scripts*, estes podem conter erros introduzidos inadvertidamente. A necessidade de adaptá-los aos eventos de acordo com o momento temporal em que se desenrolam resulta num processo demorado e complexo. As limitações inerentes à utilização deste método de simulação reduzem a qualidade dos testes a realizar por parte do operador.

Baseado na operacionalidade do sistema SIMATM, propôs-se a produção de um sistema aplicacional para a simulação de mensagens dos protocolos AFTN e OLDI. A sua concretização visa melhorar o desempenho e qualidade do processo de teste e aumentar o número de situações de teste relativamente aos actualmente executados com *scripts*.

1.6 Objectivos

A execução deste projecto visa melhorar o processo de testes das funcionalidades do sistema LISATM através do desenvolvimento de uma aplicação para simulação de mensagens do tipo AFTN e OLDI na componente de transmissão de mensagens.

Com este objectivo pretende-se desenvolver uma aplicação para simular o envio de um conjunto de mensagens para o sistema LISATM dentro de um determinado cenário de espaço aéreo.

A utilização do protótipo como ferramenta de teste servirá para teste das interfaces externas do sistema, permitindo reduzir o tempo despendido actualmente na realização de testes sobre mensagens AFTN e OLDI.

O sucesso do protótipo poderá dar origem à integração futura no sistema SIMATM, através de um módulo que possibilitará a obtenção de uma resposta mais célere ao adaptar as condições de teste e prevenir a ocorrência de erros.

1.7 Organização do documento

Este documento encontra-se organizado da seguinte forma:

Capítulo 2 – Contexto

Apresentação dos conceitos essenciais para a compreensão e realização deste projecto que servirão de base para o desenvolvimento da aplicação de simulação de mensagens.

Capítulo 3 – Metodologia e Planeamento

Apresentação do planeamento do projecto e a metodologia utilizada no seu desenvolvimento.

Capítulo 4 – Trabalho

Descrição das fases de desenvolvimento do projecto de acordo com os objectivos traçados no planeamento.

Capítulo 5 – Conclusão

Breve conclusão acerca do trabalho realizado até ao momento.

No relatório final será apresentada uma visão crítica do projecto e trabalho futuro de extensão ou melhoria do produto final.

Capítulo 2

Contexto

Este capítulo descreve os conceitos úteis para a compreensão do trabalho que servirão de base para o desenvolvimento da aplicação de simulação de mensagens.

2.1 Sistema LISATM

CONFIDENCIAL

2.2 Navegação

A navegação em espaço aéreo e terrestre rege-se pela aplicação de vastos procedimentos definidos internacionalmente, tendo como intervenientes os operadores das aeronaves que agem em concordância com o controlador aéreo. O operador da aeronave cumpre uma rota definida através de um plano de voo anteriormente submetido à entidade que regula o tráfego aéreo europeu - CFMU¹¹, que pode ser alterado à medida dos congestionamentos. O CFMU envia os planos de voos através de mensagens AFTN para o sistema LISATM, onde são processados e extraídos os dados relevantes para a FIR¹² de Lisboa.

Os principais processos de troca de informação entre CFMU, operadores de aeronaves e unidades ATC são descritos na Figura 2.

¹¹ CFMU – Control Flow Management Unit

¹² FIR – Flight Information Region

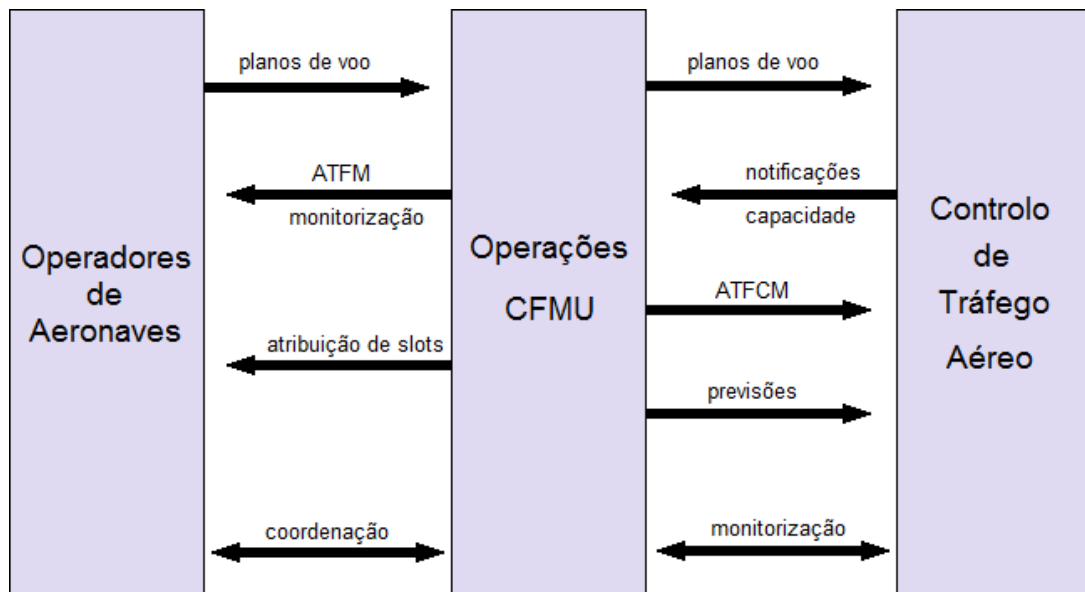


Figura 2. Estrutura operacional do CFMU.

Para além do comando das aeronaves, os operadores têm a tarefa de seguir as directrizes do controlador, mediante o cenário em causa. Para os voos que entram na FIR de Lisboa a partir de uma FIR adjacente, o órgão responsável por essa FIR notifica e coordena com o ACC de Lisboa as condições de entrada no espaço aéreo da FIR de Lisboa. Os sistemas das FIR adjacentes estão interligados com o sistema LISATM, pelo que a coordenação, notificação e revisão das condições de entrada é efectuada através da troca de mensagens OLDI. As mensagens OLDI contêm informação de notificação/coordenação e por vezes também dados da aeronave – ver Figura 3.

2.3 Aeronautical Fixed Telecommunications Network

A rede fixa de telecomunicações aeronáuticas – AFTN, administrada pela ICAO¹³ é o meio pelo qual as operações com informação relativa à aviação nacional e internacional são trocadas a nível global. A rede permite a conexão entre todos os principais aeroportos e organizações aeronáuticas através de centros nacionais/regionais onde são realizadas as comunicações de dados relativos à movimentação das aeronaves, planos de voo, informação de aeroportos, condições de meteorologia e informação

¹³ ICAO – International Civil Aviation Organization

relacionada com o controlo de tráfego aéreo. As mensagens AFTN que serão desenvolvidas para o projecto são as mensagens **FPL**, **SAM** e **CHG**. De seguida apresenta-se a finalidade de cada mensagem.

- **FPL – Flight Plan Message**

- Indicar informação sobre plano de voo da aeronave.
- Possui os seguintes campos:
 - Título mensagem;
 - Identificação da mensagem e código SSR¹⁴;
 - Regras do voo e tipo de voo;
 - Número e tipo de aeronave e categoria de turbulência;
 - Equipamento;
 - Aeródromo de partida e EOBT¹⁵;
 - Rota;
 - Aeródromo de destino, total EET¹⁶ e aeródromos alternativos;
 - Outra informação;
 - Informação suplementar;

- **CHG – Change Message**

- Indicar a modificação de plano de voo;

- **SAM – Slot Allocation Message**

- Num determinado tempo antes do EOBT em cada voo pré-alocado, é chamado o SIT¹⁷, o *slot* é alocado ao voo e uma mensagem SAM é enviada ao operador da aeronave e unidade ATC;

¹⁴ SSR – Secondary Surveillance Radar

¹⁵ EOBT – Estimated Off-Block Time

¹⁶ EET – Estimated Elapsed Time

¹⁷ SIT – Slot Issue Time

2.4 On-line Data Interchange

Quando um voo se aproxima da fronteira entre duas áreas de responsabilidade é efectuado um procedimento de notificação e coordenação de voo entre as unidades ATC para a transferência do voo.

Para garantir a segurança dos voos no processo de transferência de um serviço prestado por um ATC para outro adjacente, são trocadas informações sobre o voo através de mensagens OLDI entre os sistemas das unidades de controlo de tráfego aéreo – ver Figura 3. As mensagens OLDI que serão implementadas são **ABI**, **ACT**, **LAM** e **REV**.

ABI é uma mensagem de notificação, ACT e REV são mensagens que dizem respeito à coordenação de voos.

A mensagem LAM é uma mensagem *acknowledge* em como a mensagem de notificação/coordenação foi recebida e processada. Quando a coordenação automática falha ou é pretendida uma coordenação não normalizada (i.e. que não está conforme o acordo entre as FIR em causa), esta pode ser efectuada manualmente e neste caso não há troca de mensagens. De seguida, apresenta-se a finalidade de cada uma das mensagens envolvidas:

- **ABI – Advanced Boundary Information**

- Fornecer a aquisição de dados perdidos do plano de voo;
- Fornecer informações prévias para a próxima unidade ATC;
- Actualizar informação geral de plano de voo;
- Facilitar a correlação de tracks de radar;

- **ACT – Activate Message**

- Transmitir automaticamente detalhes de um voo proveniente de uma unidade ATC para a adjacente, antes da transferência de controlo do voo;
- Actualizar a informação de plano de voo na unidade ATC a receber o voo com a informação mais recente;

- Facilitar a distribuição e apresentação da informação de plano de voo na unidade ATC receptora nas CWP envolvidas;
- Acelerar a visualização do código CALLSIGN de correlação na unidade ATC receptora;
- Fornecer as condições de transferência para a unidade ATC receptora do voo;

- **LAM – Logical Acknowledgement Message**

- Indicar à unidade ATC transmissora, a recepção de mensagem na unidade ATC receptora;
- O processamento desta mensagem pela unidade ATC transmissora, providencia:
 - um aviso quando não é recebida esta mensagem;
 - uma indicação que a mensagem a que se refere o *acknowledge* foi recebida e processada com sucesso, livre de erros, armazenada e se relevante, está disponível para apresentação nas CWP apropriadas;

- **REV – Revision Message**

- Transmitir revisões de dados de coordenação previamente enviados numa mensagem ACT, desde que a unidade de recepção não seja alterada como resultado da modificação;

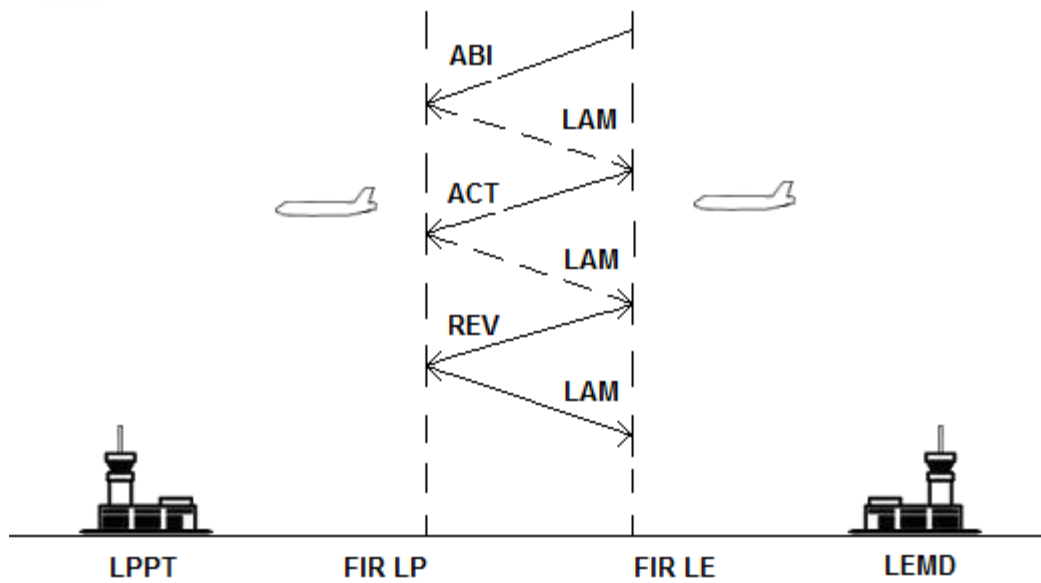


Figura 3. Exemplo de processo de transferência de controlo de voo para unidade ATC adjacente – LPPT.

Em suma, neste capítulo foram apresentados os conceitos base relacionados com o trabalho desenvolvido, em particular as mensagens do tipo AFTN e OLDI que serão objecto de simulação por parte do sistema desenvolvido para este projecto.

Capítulo 3

Metodologia e Planeamento

Neste capítulo é apresentado o planeamento do projecto e os métodos de trabalho utilizados no seu desenvolvimento.

3.1 Metodologia

Para o desenvolvimento do projecto foram tidos como base os métodos de trabalho utilizados nas sub-áreas da DSTI. Com destaque à linguagem de modelação UML que abrange todo o desenvolvimento.

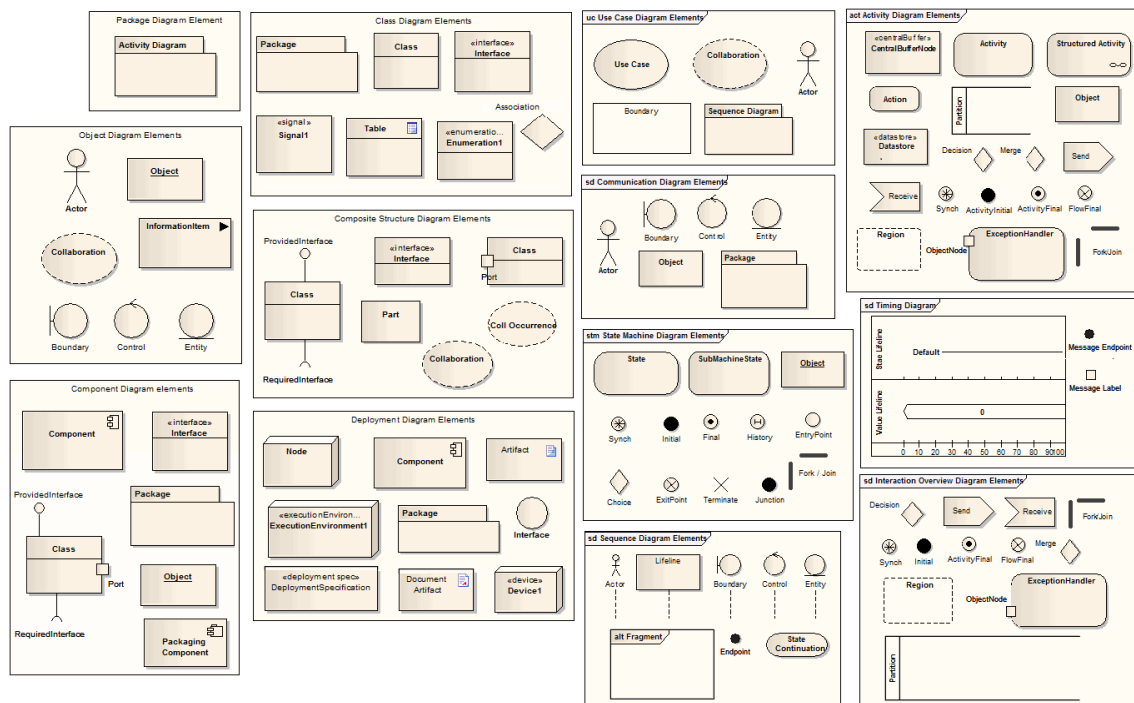


Figura 4. Exemplos de notação UML.

A linguagem **UML** – Unified Modeling Language é uma linguagem de modelação para dados orientados a objectos utilizada para especificar, construir e documentar uma vasta variedade de domínios, plataformas ou métodos – ver Figura 4.

As vantagens da linguagem residem na notação que a define, é fácil de compreender, aplicar, independentemente do processo de desenvolvimento e ao longo do todo o ciclo de vida. Permite também reutilizar e implementar os modelos de desenho através de diferentes tecnologias. Na Figura 4 é possível observar a notação UML associada a cada modelo representado em diagramas.

3.2 LISATM

CONFIDENCIAL

3.3 Processo de Desenvolvimento

Como referido no início deste capítulo, o desenvolvimento do trabalho foi seguido de acordo com a metodologia utilizada na DSTI. Como processo de desenvolvimento do trabalho, inicialmente foi adoptado o modelo cascata, no entanto verificou-se que para desenvolvimento de uma aplicação coerente com os requisitos, fiável na execução das funcionalidades e com bom desempenho seria vantajoso seguir aspectos do Processo Unificado de Desenvolvimento de Software – USDP.

O USDP, cujo ciclo de desenvolvimento é dividido em quatro fases, Concepção, Elaboração, Construção e Transição é um processo direccionado a casos de uso, centrado na arquitectura, iterativo e incremental – ver

Figura 5.

As fases são divididas numa série de iterações em que cada uma representa uma nova versão do sistema com novas funcionalidades ou melhoria delas, em relação a versões anteriores. O desenvolvimento, para além destes aspectos, centrou-se no utilizador de forma a assegurar o suporte às suas necessidades e melhoria na prática do trabalho.

Para a documentação do processo de desenvolvimento do projecto foram produzidos para a NAV Portugal os seguintes documentos do POP-20:

- DES – *Documento de Especificações de Software*
- ATD – *Acceptance Test Document* (Documento de Testes de Aceitação)

O documento de Especificações de Software - DES contém informação acerca da actividade de análise, onde são descritos e apresentados em forma de diagramas os requisitos para o sistema, modelo de casos de uso e o *modelo* de domínio. No documento de Testes de Aceitação – ATD são especificados os testes de aceitação seguindo a modelação realizada para os casos de uso. Os documentos produzidos para a NAV Portugal podem ser consultados na secção Anexos deste documento.

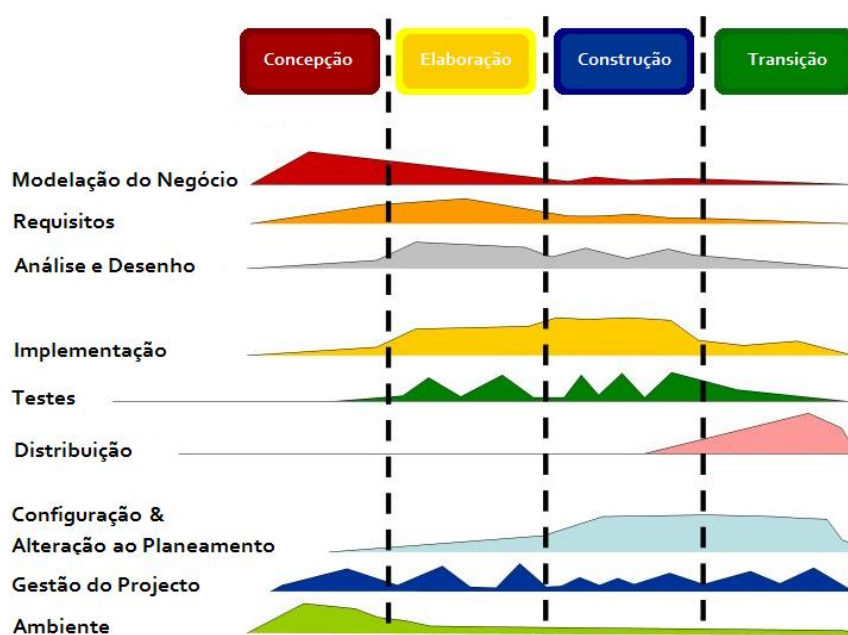


Figura 5. Ciclo de vida do Processo Unificado de Desenvolvimento de Software

Durante as fases de Desenho e Implementação foram realizados testes unitários que permitiram verificar e validar funcionalidades introduzidas na aplicação.

A fase de Testes consistiu principalmente na produção de testes de aceitação.

3.4 Planeamento do Projecto

Nesta secção é apresentado o plano de trabalho geral para a realização do projecto - Tabela 1 e o plano das tarefas associadas às actividades principais do projecto - Tabela 2. Em complemento às tarefas planeadas foi gerado o mapa de Gantt correspondente -

Figura 6.

<i>Tarefas</i>	<i>Inicio</i>	<i>Fim</i>
Formação	25-Set-2010	15-Nov-2010
Planeamento	15-Nov-2010	24-Nov-2010
Análise	25-Nov-2010	07-Jan-2011
Desenho e Codificação	10-01-2011	21-Abr-2011
Testes	24-Abr-2011	24-Jun-2011

Tabela 1. Planeamento geral de trabalho

<i>Tarefas</i>	<i>Inicio</i>	<i>Fim</i>
Formação	25-Out-2010	15-Nov-2010
Planeamento	15-Nov-2010	24-Nov-2010
produção do planeamento	15-Nov-2010	24-Nov-2010
Entregar Planeamento do Trabalho (MS Project)	24-Nov-2010	24-Nov-2010
Análise	25-Nov-2010	07-Jan-2011
captura de requisitos	25-Nov-2010	03-Dez-2010
modulação de requisitos	29-Nov-2010	07-Jan-2011
descrição casos de uso	29-Nov-2010	10-Dez-2010
diagramas casos de uso e sequência	13-Dez-2010	17-Dez-2010
modelo de domínio	20-Dez-2010	21-Dez-2010
requisitos não funcionais	28-Dez-2010	29-Dez-2010
requisitos de interface	29-Dez-2010	30-Dez-2010
produção de documento IRS e SRS (MS Word)	03-Jan-2011	07-Jan-2011
Entregar SRS (Enterprise Architecture)	07-Jan-2011	07-Jan-2011
Entregar SRS (MS Word)	07-Jan-2011	07-Jan-2011
Entregar IRS (MS Word)	07-Jan-2011	07-Jan-2011
Desenho e Codificação	10-Jan-2011	21-Abr-2011
definição da arquitectura	10-Jan-2011	12-Jan-2011
descrição da arquitectura	10-Jan-2011	12-Jan-2011
desenhar módulos	13-Jan-2011	03-Fev-2011

definir modelo de desenho	13-Jan-2011	17-Jan-2011
diagramas de classes	18-Jan-2011	24-Jan-2011
desenho do modelo conceptual (ERD)	25-Jan-2011	03-Fev-2011
produção de código	04-Fev-2011	04-Abr-2011
testes unitários	11-Abr-2011	21- Abr-2011
realizar testes unitários	11-Abr-2011	15- Abr-2011
efectuar e testar alterações ao código	18-Abr-2011	21- Abr-2011
Entregar Software Delivery (JAVA)	21-Abr-2011	21- Abr-2011
Testes	02-Mai-2011	24-Jun-2011
produção de especificações de teste	02-Mai-2011	10-Mai-2011
produção de documento Enterprise Architecture	14-Jun-2011	20-Jun-2011
realizar testes de integração	16-Mai-2011	03-Jun-2011
realizar testes de sistema/aceitação	01-Jun-2011	13-Jun-2011
actualizar documentos	14-Jun-2011	16-Jun-2011
produção de relatório de testes	17-Jun-2011	24-Jun-2011
Entregar Especificações de Teste (Enterprise Architecture)	24-Jun-2011	24-Jun-2011
Entregar Relatório de Testes	24-Jun-2011	24-Jun-2011

Tabela 2. Planeamento detalhado de trabalho

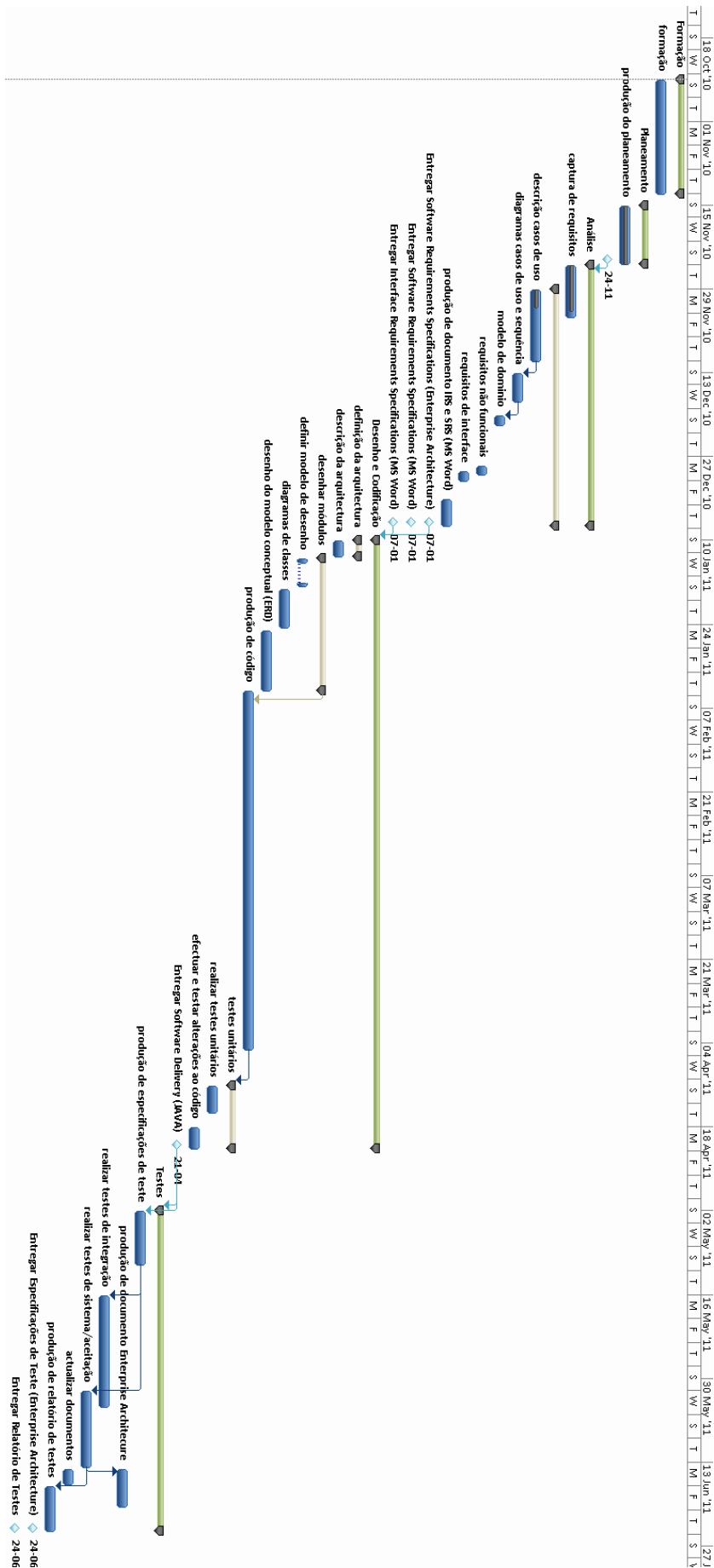


Figura 6. Mapa de Gantt referente ao planeamento do projecto

3.4.1 Desvios

Durante a execução de algumas tarefas definidas no planeamento surgiu a necessidade de prolongar o período de tempo agendado. Sempre que se verificou esta situação procurou-se ajustar o tempo exigido à conclusão da tarefa, em conformidade com o período de tempo agendado em tarefas adjacentes e suas dependências. Uma das soluções encontradas para fazer face às exigências do projecto, sem prejudicar o tempo de desenvolvimento foi estender o tempo de trabalho laboral definido na empresa.

A definição do período de tempo a atribuir às tarefas a realizar no planeamento, revelou-se particularmente difícil. A dificuldade surgiu na falta de conhecimento profundo sobre o negócio enquadrado no projecto a desenvolver, tornando-se complicado definir um tempo legítimo para a concretização de algumas tarefas. Nomeadamente na fase de análise e desenho da aplicação.

Na fase de análise, a modulação de requisitos foi a tarefa que absorveu mais tempo.

Fruto do processo de descrição de casos de uso, foram desenvolvidas várias versões de um documento de descrição de casos de uso. Esse documento foi construído através da captura de requisitos que teve como base um documento de requisitos relativo ao produto a desenvolver.

A finalização do documento de descrição de casos de uso, surgiu após várias reuniões de intervenção e esclarecimento por parte das pessoas responsáveis pelo acompanhamento do desenvolvimento da aplicação. O tempo demorado na descrição de casos de uso, relativamente ao enquadramento técnico do processo envolvente, nos conceitos técnicos aplicados no centro de custo SISINT e aos requisitos exigidos pelos técnicos envolvidos, motivou o prolongamento do tempo agendado para a tarefa.

Para a produção dos documentos de qualidade de software foram utilizadas as ferramentas MS Word, para escrita de documentos, e a ferramenta Enterprise Architecture, para modulação. Para produzir o documento DES com a informação modelada nos diferentes diagramas da fase de análise foi utilizada a função de criação de documentos da ferramenta EA.

Com o objectivo de aproveitar funcionalidades fornecidas pela ferramenta, juntamente com a inexistência de um template adaptado a essas mesmas funcionalidades, foi necessária a configuração de um novo template DES. A configuração do template consistiu na formatação baseada em templates existentes e na organização da informação a incluir no documento. Como esta tarefa não fazia parte do planeamento, esta originou um desvio no tempo para as tarefas agendadas sem implicações na concretização atempada das tarefas subsequentes.

Capítulo 4

Desenvolvimento do Projecto

Neste capítulo é descrito o trabalho produzido em todas as fases de desenvolvimento do projecto. A estrutura dos tópicos apresentados corresponde às principais actividades do processo de desenvolvimento. Em cada tópico, caso se tenha verificado, para além da descrição do trabalho desenvolvido serão referidas as decisões tomadas, dificuldades encontradas e suas soluções.

Todo o trabalho desenvolvido durante as fases de Análise, Desenho e Testes foi modelado com o auxílio da ferramenta Enterprise Architecture e descrito nos documentos DES e ATD gerados através da mesma. Os documentos estão disponíveis na secção Anexos deste relatório.

4.1 Análise de Requisitos

CONFIDENCIAL

4.2 Desenho

Nesta fase do trabalho foram definidos a arquitectura do sistema e o desenho de *software*. O desenho de *software* é apresentado sob a forma de um modelo de classes. A arquitectura do sistema foi definida com base na arquitectura física e lógica do sistema, representada sob a forma de um diagrama de distribuição, de componentes e pacotes.

4.2.1 Padrões de Desenho

O desenho de *software* da aplicação foi definido de acordo com o padrão de desenho *Model-View-Controller*.

O padrão de desenho *Model-View-Controller* (MVC) consiste em três tipos de objectos. O *Model* é o objecto da aplicação e implementa o padrão *Observer*, a *View* é responsável pelos aspectos visuais do ecrã e o *Controller* define a maneira como a interface de utilizador actua mediante os comandos de entrada. A utilização do padrão de desenho *Observer* permite anexar múltiplas *Views* a um *Model* para fornecer diferentes apresentações e mantém o *Model* independente do *Controller* e da *View*.

O funcionamento do padrão MVC ilustrado na Figura 7 é o seguinte:

- O utilizador interage com a *View* (por exemplo, *click* de um botão), esta avisa o *Controller* sobre a ocorrência da acção;
- O *Controller* de acordo com acção na *View* pede ao *Model* para mudar o estado (por exemplo, após o *click*, o *Controller* interpreta o significado da acção e decide como o *Model* deve ser manipulado. O *Controller* pode pedir à *View* para mudar a sua apresentação;
- Sempre que o *Model* é modificado, este notifica a *View* que depende dele.
- A *View* consulta o *Model* sobre o seu novo estado, que pode ser requerido pelo *Controller* ou pelo aviso do *Model* perante uma alteração.
- A *View* tem de garantir que reflecte exactamente o estado do *Model*.

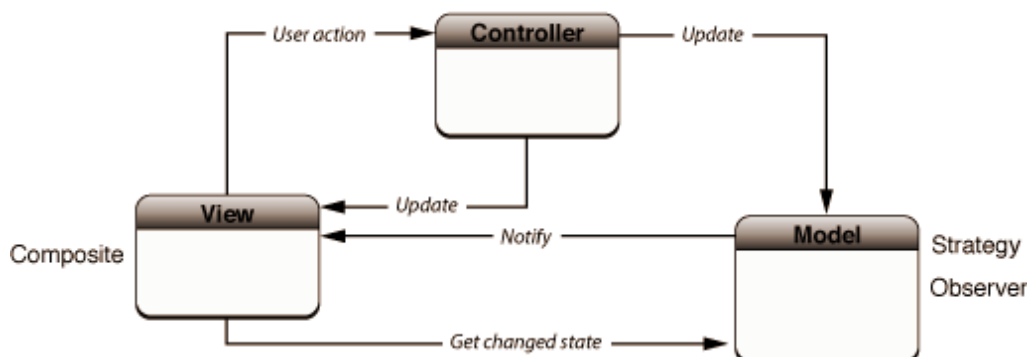


Figura 7. Estrutura do padrão *Model-View-Controller* para uma aplicação *GUI*

Com o objectivo de facilitar a organização, edição, formatação e tratamento de informação da aplicação a implementar foram analisados alguns padrões de desenho. De forma a garantir a boa implementação das premissas mencionadas foram tidos como base os padrões de desenho *Strategy*, *Factory Method*, *Decorator*, *Observer* e *Composite*.

O padrão *Strategy* é utilizado para o modelo – *Model*, o qual possibilita o acesso a diferentes tipos de dados (cenário, voo, mensagem). A classe DAO - *Data Access Object* é implementada com base neste padrão.

O padrão Composite é aplicado na *View*. O sistema ao possuir diferentes *View* necessita por vezes que actuem de forma similar, o padrão permite integrar vários objectos possibilitando ao sistema manipulá-los como se fossem somente um único.

O padrão Factory Method permite a criação de objectos sem especificar o tipo de classe do objecto, o que torna útil na especificação de objectos de subclasses. A classe Mensagem é um exemplo dessa utilização, na criação das mensagens ABI, ACT, REV, CHG e SAM.

O padrão Decorator é utilizado nas *Views*, mais concretamente na implementação de elementos de apresentação comuns, principalmente em informação de mensagem.

4.2.2 Modelo de Classes

CONFIDENCIAL

4.3 Implementação

A implementação do projecto consiste essencialmente em satisfazer os requisitos na forma como foram especificados na fase de desenho através de programação de código Java. O documento DES produzido serve para assegurar que a implementação satisfaça os requisitos detalhados.

A tarefa de programação cumpriu várias etapas durante a sua execução, como a escrita de código, a verificação do mesmo através de testes unitários e solução para os casos de ocorrência de erro ou falha na solução implementada. A linguagem de programação Java foi a seleccionada para o desenvolvimento da aplicação. Para o desenvolvimento do código foram tidos em conta os requisitos da aplicação, a construção da interface com o utilizador através da utilização das Java Foundation Classes (JFC) Swing, a experiência adquirida em outros projectos e a arquitectura dos sistemas a integrar no futuro. Para assegurar uma futura integração nos sistemas operacionais da NAV Portugal, o código fonte desenvolvido para o projecto foi compilado na versão 1.4.2 do Java Runtime Environment (JRE), garantindo a compatibilidade com o sistema operativo OS/2.

Para a realização da tarefa de codificação, foi escolhida a ferramenta Eclipse, devido às características funcionais e experiência pessoal adquirida em projectos académicos.

Para a concretização do modelo de dados relacionais da aplicação, recorreu-se à ferramenta de gestão de base de dados MySQL. A opção por esta ferramenta baseou-se

no facto de possuir licença aberta, com suporte através da comunidade online, oferecer ligação com a plataforma Java, capacidade de adaptação a qualquer sistema e garantir o desempenho desejado para a aplicação.

Para a implementação do programa aplicacional foram consideradas premissas importantes e prioritárias, a manutenção, reutilização e extensão. Com base nelas a arquitectura da aplicação foi implementada de acordo com o padrão de desenho Model-View-Controller explicado anteriormente.

A utilização de métricas para o código não foi considerada pois não traria benefícios para o desenvolvimento das estruturas de código. As boas noções da prática de programação e de complexidade ciclomática de instruções ajudaram nesse processo. A sua inclusão no processo de programação teria impacto considerável no planeamento definido para esta fase do projecto.

4.3.1 Modelo de Dados

CONFIDENCIAL

4.3.2 Estrutura do Código

CONFIDENCIAL

4.3.3 Comunicação

CONFIDENCIAL

4.3.4 Interface de Utilizador

A concepção da interface do utilizador foi baseada nos esboços produzidos anteriormente na tarefa de prototipagem e a partir de princípios de desenvolvimento, dos quais se destacam as regras de ouro de Ben Shneiderman:

1. Esforçar-se por manter a consistência;
2. Permitir o uso de atalhos para os utilizadores mais frequentes;
3. Oferecer retorno informativo;

4. Conceptualizar sequência de acções de modo a fornecer oclusão;
5. Oferecer prevenção e fácil tratamento de erros;
6. Facilitar o retrocesso de acções;
7. Fornecer a sensação de controlo ao utilizador;
8. Reduzir a carga de memória a curto prazo;

Para inspecção da interface durante a sua concepção foi utilizado um método de avaliação informal que teve como referência os princípios heurísticos de usabilidade. As heurísticas de Nielsen foram tidas como base dessa avaliação. Eis a lista delas:

1. Informar continuamente o utilizador das acções tomadas – *status* do sistema;
2. Equivalência entre o sistema e o mundo real, aplicar a linguagem do utilizador;
3. Controlo e sensação de liberdade por parte do utilizador;
4. Consistência e padrões;
5. Prevenção de erros;
6. Reconhecer em vez de relembrar;
7. Flexibilidade e eficiência – utilização de atalhos;
8. Estética e desenho minimalista;
9. Auxílio ao utilizador no reconhecimento, diagnóstico e recuperação de erros;
10. Ajuda e documentação;

Neste campo da usabilidade, o desenho da interface foi concebido de forma a assegurar que a aplicação fosse adequada para as tarefas a desempenhar, fácil de utilizar e adaptado à experiência e conhecimento dos utilizadores. Entre os factores de usabilidade aplicados na interface, o factor visibilidade consistiu em apresentar de forma visível as partes relevantes da aplicação e as acções que o utilizador pode efectuar.

A compreensão natural do que os controlos pretendem executar foi trabalhada de forma a garantir o correcto mapeamento.

O retorno - *feedback* foi realizado de maneira a que o utilizador se soubesse situar no programa, de acordo com as acções tomadas.

Os estilos de interacção foram definidos de acordo com a experiência e o modo como os utilizadores interagem com a aplicação. Para tal, a interacção pode ser efectuada com recurso a menus, formulários e botões. Os menus (*Pull Down* e encadeados) e suas opções simplificam a interacção aos utilizadores podendo ainda ser acedidos através de atalhos, enquanto os formulários possibilitam um estilo orientado para a informação com um esquema bem definido que permite a introdução simples de dados.

De seguida são apresentados exemplos da estrutura da interface de utilizador da aplicação recorrendo a *printscreens*. Os exemplos são apresentados segundo os tópicos Autenticação, Gestão de Cenários, Gestão de Voos, Gestão de Mensagens e Simulador. Esta organização pretende relacionar os tópicos definidos com os casos de uso Registrar Utilizador, Gerir Cenários, Gerir Voos, Gerir Mensagens e Gerir Exercício, respectivamente.

- **Autenticação** - Na janela de abertura - Figura 8, o utilizador selecciona a aplicação a executar (Gestor ou Simulador) após a autenticação;

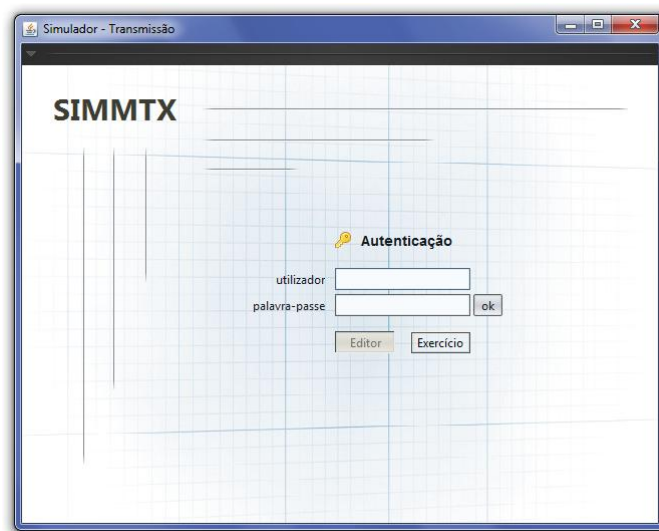


Figura 8. Interface de utilizador – Autenticar utilizador

- **Gestão de Cenários** - Na Figura 9, a interface disponibiliza uma lista de cenários para selecção e opções de gestão (Novo Cenário/Sair). A Figura 10 apresenta o estado de um cenário após a selecção por parte do utilizador e janelas correspondentes à selecção das opções Copiar e Apagar

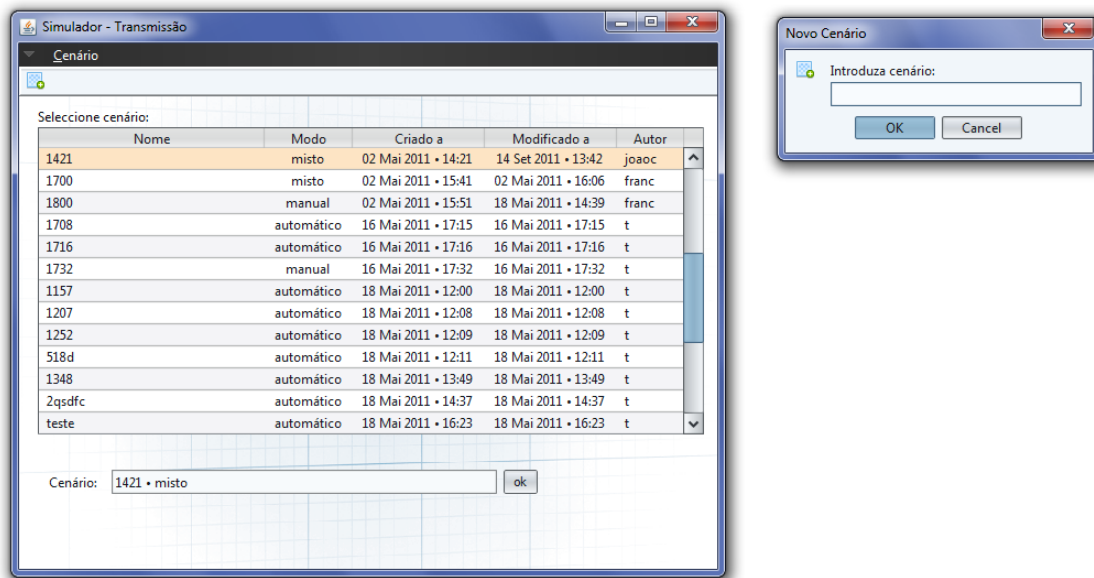


Figura 9. Gestor de cenários – Seleccionar cenário; Introduzir nome de cenário

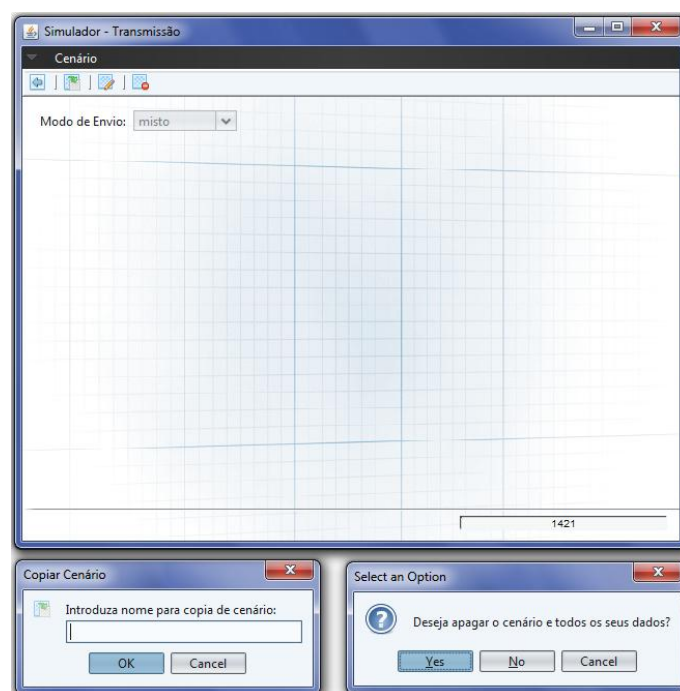


Figura 10. Gestor de cenários – Editar cenário; Copiar cenário, Apagar cenário

- **Gestor de Voos** - Na Figura 11, a interface disponibiliza uma lista de voos para selecção e opções de gestão (Novo Voo/Sair). A Figura 12 apresenta a edição de um voo e janelas correspondentes à selecção das opções Copiar e Apagar e opções no menu do estilo *pull down*

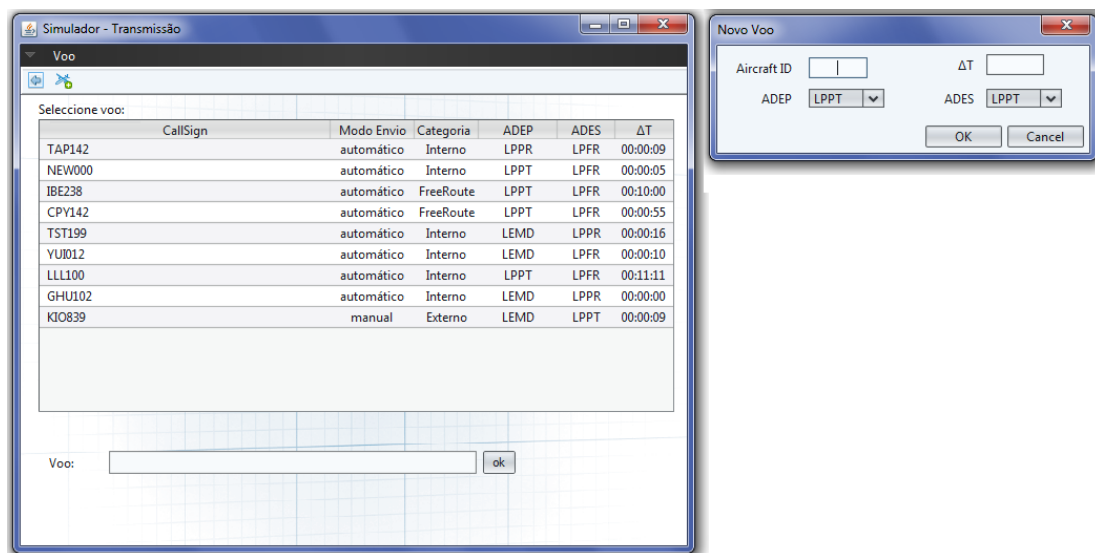


Figura 11. Gestor de voos – Seleccionar voo; Criar novo voo

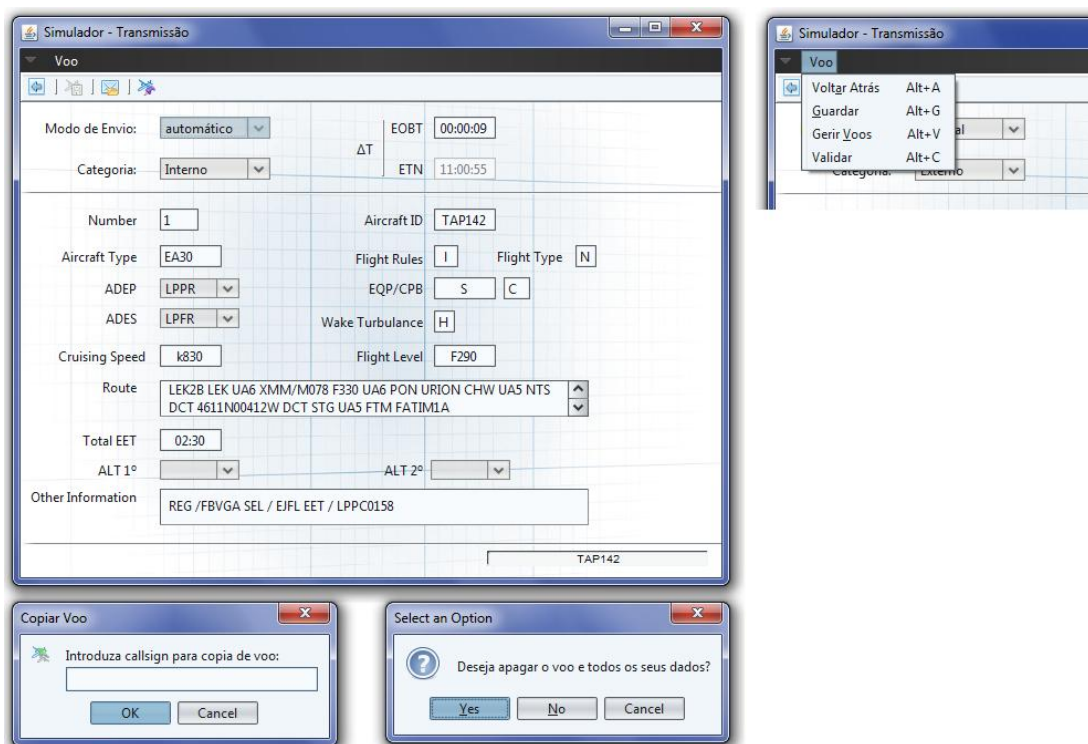


Figura 12. Gestor de Voos – Editar voo; Copiar voo, Apagar voo; Menu *Pull Down*

- **Gestor de Mensagens** - Na Figura 13 a interface disponibiliza uma lista de mensagens para selecção e opções de gestão (Nova Mensagem/Sair). A Figura 14 apresenta a edição de uma mensagem, validação de informação e visualização da sintaxe de mensagem, juntamente com as janelas correspondentes à selecção das opções Validar Mensagem e Ver Sintaxe

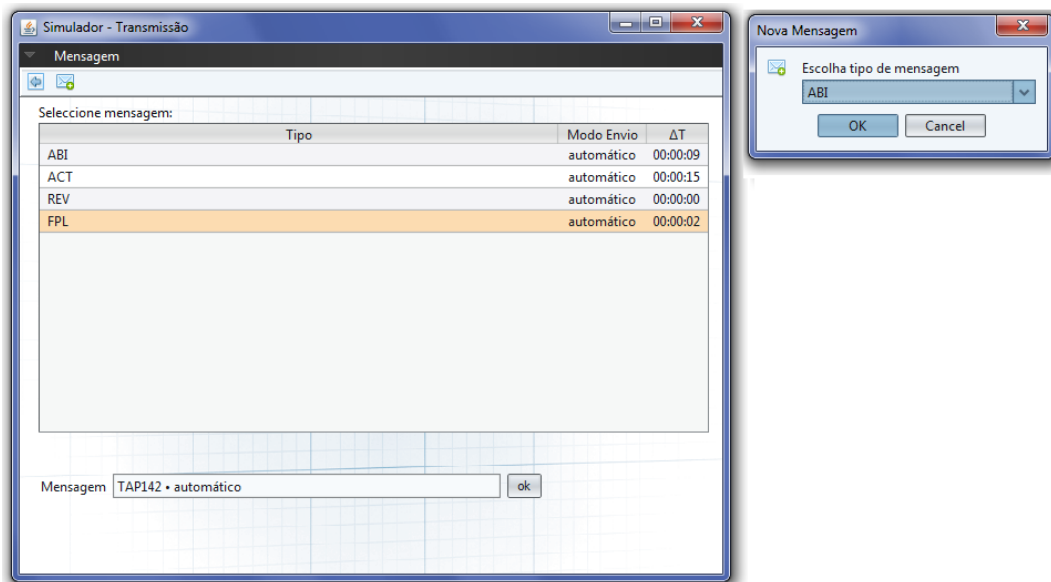


Figura 13. Gestor de Mensagens – Seleccionar mensagem, Criar nova mensagem

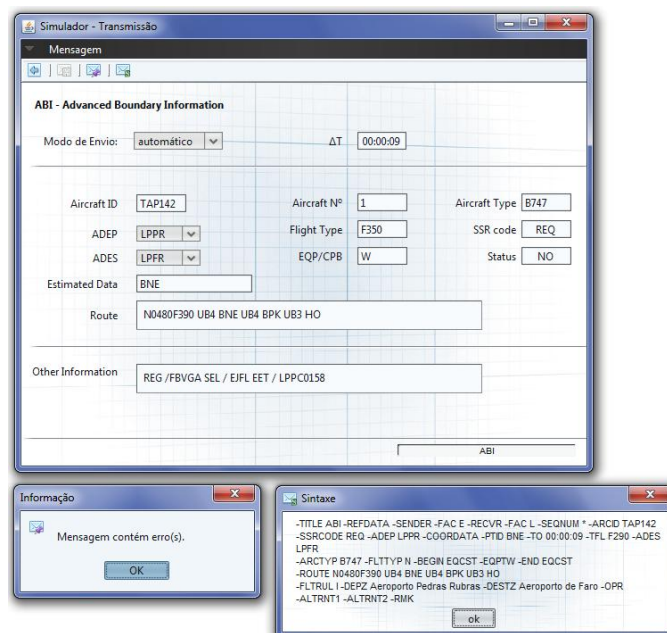


Figura 14. Gestor de Mensagens – Editar mensagem; Validar mensagem; Ver Sintaxe

- **Simulador** - A Figura 15 apresenta o estado em que o utilizador acede ao exercício de simulação com informação de um cenário seleccionado previamente e o diálogo de gestão de voos do exercício, após a selecção da opção Gerir Voos. Na Figura 16 é apresentado um exemplo da execução de um exercício de simulação de mensagens. No exemplo, observa-se que o tempo de execução regista 10 segundos decorridos e a tabela de eventos é actualizada. À medida que o tempo de envio de mensagem (deltaT) coincide com o tempo de execução do exercício é enviada a mensagem. Comparando as duas figuras correspondentes ao simulador verifica-se que o modo de envio das mensagens encontra-se inactivo antes do *play* do exercício.

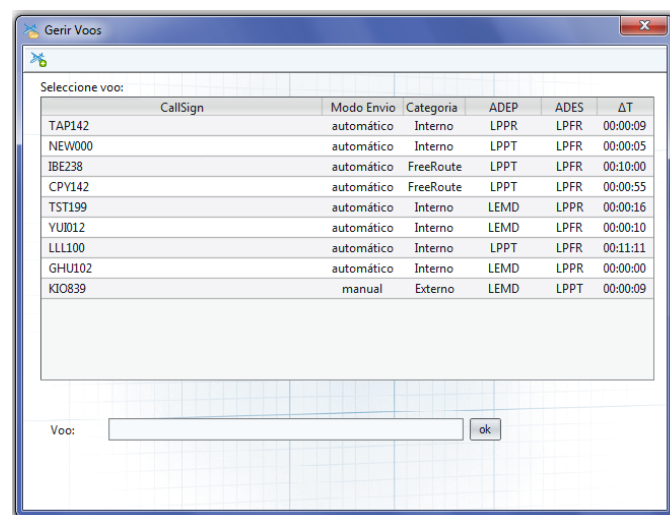


Figura 15. Simulador – Gerir mensagens do exercício

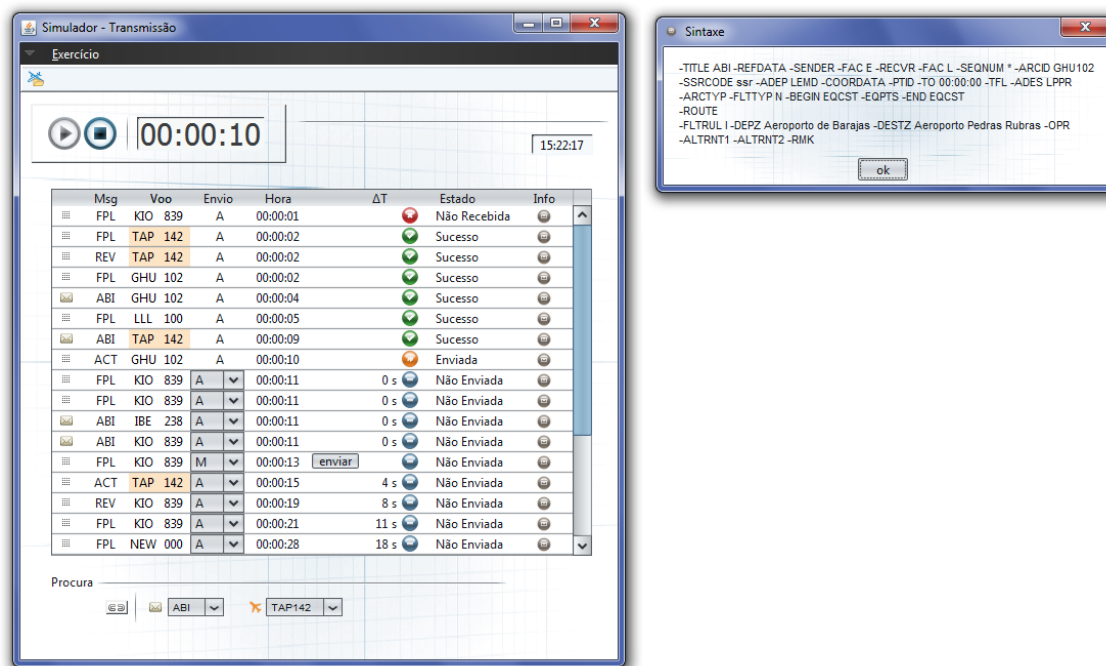


Figura 16. Simulador – Iniciar exercício

Como se pode observar em todas as figuras apresentadas é permitido ao utilizador a interacção com as componentes de interface, de forma directa e com a capacidade - *affordance* desejável para o tipo de informação a apresentar.

Todos os aspectos visuais da interface foram produzidos de raiz para a aplicação, nomeadamente, imagens de fundo e ícones, à excepção da formatação *standard* de componentes Swing aplicada pela framework *substance* utilizada como *Look And Feel* da aplicação.

4.4 Testes

A concepção de testes tem o objectivo principal de encontrar erros, ao invés de demonstrar que o sistema funciona na plenitude das suas funcionalidades. Os testes devem testar que os requisitos de sistema e utilizador são cumpridos, os quais devem ser realizados por uma pessoa independente à codificação do projecto, o que não foi o caso devido à falta de elementos disponíveis na secção do departamento para realizar essa tarefa.

Ao longo do processo desenvolvimento, a implementação de novas funcionalidades devem ser verificadas e validadas através de testes unitários e posteriormente por testes de integração, testes de sistema (informais) e de aceitação.

Os testes unitários, também denominados de caixa preta, são testes funcionais e foram feitos durante a actividade de programação, no entanto sem recorrer a

ferramentas direccionadas a essa função, como o JUnit, por não estar no âmbito do projecto e no planeamento.

Os testes de integração não foram realizados devido a não ter sido possível integrar a parte de comunicação entre a aplicação e o sistema LISATM. Inerente a esta situação, os testes ao sistema relacionados com testes de capacidade e desempenho geral do sistema foram informalmente verificados durante a implementação da comunicação entre a aplicação com a interface XATMI.

Para além destes testes mencionados foram realizados outro tipo de testes durante a implementação do sistema, que consistiram essencialmente em testes à interface de utilizador para verificação de incoerências, erros ou falhas. A realização destes testes à interface contribuiu fortemente para o desenvolvimento de uma interface coerente, intuitiva e funcional que resultou posteriormente numa melhor avaliação do retorno dos casos de teste.

Os testes nesta fase incidiram maioritariamente no planeamento e execução de testes de aceitação, os quais são apresentados de seguida através do modelo de testes.

Os testes de aceitação são testes formais, conduzidos para determinar se um sistema satisfaz ou não os critérios de aceitação correspondentes, permitindo ao cliente/utilizador aceitar ou não o sistema.

Como complemento foram gerados automaticamente ainda testes de cenário dos casos de uso através da ferramenta Enterprise Architecture. Os testes realizados podem ser consultados no documento **ATD** na secção Anexos.

4.4.1 Modelo de Testes

CONDIFENCIAL

Capítulo 5

Conclusão

A NAV Portugal com o sistema de gestão de tráfego aéreo - LISATM oferece mecanismos de controlo e gestão de informação em tempo-real aos controladores para auxílio das suas funções. Para fazer face aos requisitos operacionais do sistema LISATM é essencial a execução de testes e validação de funcionalidades antes da sua integração. Os testes ao sistema com mensagens do tipo AFTN e OLDI efectuados através da injeção de scripts são lentos e difíceis de executar devido à complexidade do processo manual que acarretam. Para superar as limitações do método utilizado foi desenvolvido um sistema aplicacional que permite a injeção, criação e gestão de informação das mensagens dos protocolos AFTN e OLDI. O objectivo da aplicação é tornar mais fácil, rápido e automáticos os testes de funcionalidades do sistema LISATM.

Para a concretização da aplicação foi seguido um processo de desenvolvimento que teve como base os métodos de trabalho utilizados na DSTI, em particular na subárea SISINT. O processo de desenvolvimento incluiu as fases de Análise de Requisitos, Desenho, Implementação e Testes e cujos resultados apresentados neste relatório permitem-me concluir que a aplicação desenvolvida cumpre os requisitos necessários para a concretização da simulação de mensagens AFTN e OLDI.

O objectivo do projecto foi cumprido, o que contribui para a melhoria do processo de testes de funcionalidades do sistema de gestão de tráfego aéreo LISATM.

Como análise técnica do trabalho, esta deve começar com uma viabilidade técnica do sistema, o que no caso deste projecto se verificou existir implicitamente pois houve acesso a todas as ferramentas necessárias para a análise e codificação.

A componente de formação foi essencial para obter uma aprendizagem consolidada do conhecimento exigido para o projecto e execução das tarefas realizadas.

Como *`errare humanum est'*, ocorreram alguns desvios no planeamento, essencialmente em actividades de análise e recodificação de código fonte, mas que contribuíram positivamente em termos de aprendizagem para futuros desafios. No geral, foram cumpridas com sucesso as tarefas do planeamento delineado para o projecto e atingidos os objectivos propostos.

Todos os desafios inerentes às actividades desenvolvidas durante o projecto contribuíram para consolidar o conhecimento adquirido ao longo do percurso académico e profissional até ao momento. O estágio foi assim importante para aplicar o conhecimento adquirido na área de especialização em sistemas de informação e outras áreas da informática relacionadas com o projecto, como a engenharia de *software* e arquitectura de redes.

O projecto para além de me dar a conhecer o processo de negócio em que se insere deu-me ainda a possibilidade de concretizar o modelo de desenvolvimento aplicado na Direcção de Sistemas de Tecnologias de Informação da empresa NAV Portugal.

Há a referir que existiu em paralelo a este projecto o desenvolvimento de outro projecto que tem o objectivo de simular mensagens AFTN e OLDI para a vertente de Recepção de mensagens.

O sucesso destes projectos de simulação de mensagens AFTN e OLDI poderão através da integração futura de um módulo possibilitar a obtenção de uma resposta mais célere na adaptação das condições de teste e prevenção na ocorrência de erros.

É com elevado grau de satisfação que dou por concluído o relatório de estágio do projecto proposto e realizado.

Bibliografia

1. “Procedures for Air Navigation Services – Air Traffic Management”, DOC 4444 ATM/501, 15ª Edition, 2007, ICAO
2. “Eurocontrol Standard Document for On-Line Data Interchange (OLDI)”, 2001, Edition 2.3, Eurocontrol,
http://www.eurocontrol.int/oldi/gallery/content/public/doc/oldi_3-00.pdf
3. “General & CFMU Systems”, 18-03-2010, Eurocontrol
4. “Integrated Initial Flight Plan Processing System - IFPS Users Manual”, Edition 14.1, 7-04-2010, Eurocontrol
5. Francisco Caldeira, Rui Azedo, “Documento de Especificações de Sistema – TWRATM Porto – Processamento Inicial de Planos de Voo”, 20-09-2010, NAV Portugal, E.P.E.
6. Manuel Dias, “SIMATM – Artigo Navegar - Management”, 11-08-2006, NAV Portugal, E.P.E.
7. Francisco Caldeira, Rui Azedo, “Documento de Especificações de Sistema – TWRATM Porto – Coordenação -Entrada na RIV”, 20-09-2010, NAV Portugal, E.P.E.
8. Rui Azedo, Carlos Santos, Francisco Caldeira, José Vermelhudo, Maria Serrano, “Flight Data Processing System – Functional Requirements Specifications – General”, 19-03-2010, NAV Portugal, E.P.E.

9. Peter Chen, The Entity-Relationship Model – Toward a Unified View of Data, International Conference on Very Large Data Bases - Framingham, Massachusetts, Sept. 22-24, 1975, Massachusetts Institute of Technology
10. Model-View-Controller – MSDN Library,
<http://msdn.microsoft.com/en-us/library/ms978748.aspx>
11. Web Site ICAO – International Civil Aviation Organization,
<http://www.icao.int/>
12. Qualidade POP-20 – Prestação de Serviços de Desenvolvimento de Sistemas (Apresentação PowerPoint), NAV Portugal
13. XMLRAC – xml Register Alive Connection, Interface Design Document, versão 1.0, 2002, NAV Portugal
14. TortoiseSVN – Subversion Client Control System,
<http://tortoisesvn.tigris.org>
15. Best Practice Software Engineering, 06-04-2011, Quality-Software-Engineering group of the Vienna University of Technology,
<http://best-practice-software-engineering.ifs.tuwien.ac.at/>
16. Molina, H. G., Ullman, J. D. and Widom, J., Database Systems - The Complete Book, Prentice Hall, 2002.
17. Creating a Custom Look and Feel, Oracle Sun Developer Network,
<http://java.sun.com/products/jfc/tsc/articles/sce/index.html>
18. Design Guidelines, Java Sun Microsystems Inc., 1999,
<http://java.sun.com/products/jlf/ed1/dg/higa.htm>
19. Design Principles, Mark D. Huang, 23-02-1997,
http://www.cc.gatech.edu/classes/cs6751_97_winter/Topics/design-princ/

20. Recommendations for HMI Evaluation in the context of CWP Development:
A Synthesis of Relevant Literature, 2003, Eurocontrol Experimental Center,
http://www.eurocontrol.int/eec/gallery/content/public/document/eec/report/2003/003_Recommendations_for_HMI_Evaluation.pdf
21. Gamma, Erich; Helm, Richard; Ralph, Johnson; John, Vlissides, Design
Patterns - Elements of Reusable Object-Oriented Software, 1997, Addison
Wesley
22. Martin Fowler, Development of Further Patterns of Enterprise Application
Architecture, 2011,
<http://martinfowler.com/eaDev/index.html>
23. UML Tutorial, Enterprise Architecture, 2011,
<http://www.sparxsystems.com/uml-tutorial.html>

Glossário

ABI - Advance Boundary Information Message

ACC - Area Control Centre

ACT Activate Message

ADEXP - ATS Data Exchange Presentation

AFTN – Aeronautical Fixed Telecommunications Network

AOS - Airport Operational Services

ARTAS - ATM surveillance (Radar) Tracker And Server

ASD - Air Situation Display

ASN1 - Abstract Syntax Notation 1

ATC - Air Traffic Control

ATCC - Air Traffic Control Center

ATIS - Automated Terminal Information Service

ATM - Air Traffic Management

ATO - Actual Time Over

ATS – Air Traffic Service

CALLSIGN - Aircraft Identification

CFMU - Central Flow Management Unit

COP - Co-ordination Point

COPN - Entry Co- ordination Point

COPX - Exit Co-ordination Point

CWP – Control Working Position

DSTI – Direcção de Sistemas e Tecnologias de Informação

EET – Estimated Elapsed Time

EID - Electronic Information Distribution

ETN - Estimated Time Over COPN

ETX - Estimated Time Over COPX

EOBT – Estimated Off-Block Time

EUROCONTROL – European Organization for the Safety of Air Navigation

FDPS - Flight Data Processing System

FIR – Flight Information Region

GUI - Graphical User Interface

ICAO - International Civil Aviation Organization

IFPS - Initial Flight Plan Processing System

LAM - Logical Acknowledgement Message

LEMD – código ICAO do aeroporto de Barajas, Madrid

LPPT – código ICAO do aeroporto da Portela

LISATM – Lisbon FIR ATM System

MVC - Model View Controller

NAV - Navegação Aérea de Portugal

ODS - Operator Display System

OLDI - On-Line Data Interchange

POACCS - Portuguese Operational Air traffic Control Center System

Pre-OJT – Pre-On Job Training

REV – Revision Message

SIMATM – Simulator ATM System

SISINT - NAV-DSTI: Sistemas de Interface com o utilizador

SISLOG - NAV-DSTI: Sistemas de Logística

SISPRO - NAV-DSTI: Sistemas de Produção

SISQUA - NAV-DSTI: Sistemas de Gestão da Qualidade e Safety

SSR - Secondary Surveillance Radar

TWRATM – Control Tower ATM System

UDP - User Datagram Protocol

URD – User Requirements Document

XML - Extensible Markup Language

XMLRAC - Xml Register-Alive Connection